



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/795,923	03/08/2004	Daniel Kegel	I004-P03074US	1818

33356 7590 02/28/2012
SoCAL IP LAW GROUP LLP
310 N. WESTLAKE BLVD. STE 120
WESTLAKE VILLAGE, CA 91362

EXAMINER

JEAN GILLES, JUDE

ART UNIT	PAPER NUMBER
----------	--------------

2447

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

02/28/2012

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

USPTO@socalip.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte DANIEL KEGEL

Appeal 2009-010841
Application 10/795,923¹
Technology Center 2400

Before JOSEPH L. DIXON, HOWARD B. BLANKENSHIP, and
JAMES R. HUGHES, *Administrative Patent Judges*.

HUGHES, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellant appeals under authority of 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-35. The Board of Patent Appeals and Interferences (BPAI) has jurisdiction under 35 U.S.C. § 6(b).

We affirm.

¹ Application filed March 8, 2004. The real party in interest is Ixia. (App. Br. 1.)

Appellant's Invention

The invention at issue on appeal concerns a telecommunication network testing system, machine-readable medium, and method of creating network traffic replicating activities. The method includes steps of receiving a test script including a sequence of commands or instructions and invoking a script interpreter. The commands cause network traffic to be created and transmitted by the network testing system. The method then launches an application thread to execute the test script. The application thread in turn launches a protocol engine for each of the commands in the test script such that each protocol engine has an associated command and invokes a call to the operating system to accomplish and execute the command. (Spec. ¶¶ [0027]-[0029], [0032]-[0034]; Abstract.)²

Representative Claim

Independent claim 1, reproduced below with the key disputed limitations emphasized, further illustrates the invention:

1. A method of creating network traffic replicating activities of a large number of users comprising:
 - receiving a test script including a plurality of commands
 - invoking a script interpreter
 - launching an application thread to execute the test script
 - invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command,

² We refer to Appellant's Specification ("Spec."); Appeal Brief ("App. Br.") filed August 6, 2008; and Reply Brief ("Reply Br.") filed December 22, 2008. We also refer to the Examiner's Answer ("Ans.") mailed November 13, 2008.

each protocol engine executing its associated command.

References

The Examiner relies on the following references as evidence in support of the rejection:

Smith	US 6,091,802	July 18, 2000
Jameson	US 2003/0107596 A1	Jun. 12, 2003
Averbuj	US 2005/0257109 A1	Nov. 17, 2005 (filed Jul. 29, 2003)

Rejections on Appeal

1. The Examiner rejects claims 1, 2, 5-12, 15-23, 26-28, and 31-33 under 35 U.S.C. § 103(a) as being unpatentable over the combination of Smith and Averbuj.

2. The Examiner rejects claims 3, 4, 13, 14, 24, 25, 29, 30, 34, and 35 under 35 U.S.C. § 103(a) as being unpatentable over the combination of Smith, Averbuj, and Jameson.

ISSUES

Based on our review of the administrative record, Appellant's contentions, and the Examiner's findings and conclusions, the pivotal issues before us are as follows:

1. Does the Examiner err in combining Smith and Averbuj and concluding that the combination would have collectively taught or fairly suggested "invoking a protocol engine for each of the commands in the test

script such that each protocol engine has an associated command, [and] each protocol engine execut[es] its associated command” as recited in claim 1?

2. Does the Examiner err in concluding that the combination of Smith and Averbuj would have collectively taught or fairly suggested an “application thread” as recited in claim 1?

3. Does the Examiner err in concluding that the combination of Smith and Averbuj would have collectively taught or fairly suggested that “the commands in the test script simulate actions taken by a network user” as recited in claim 2?

4. Does the Examiner err in concluding that the combination of Smith and Averbuj would have collectively taught or fairly suggested that “the test script causes network traffic to be produced” as recited in claim 5?

5. Does the Examiner err in concluding that the combination of Smith and Averbuj would have collectively taught or fairly suggested the “user space” and “operating system space” as recited in claims 21, 26, and 31?

FINDINGS OF FACT

We adopt the Examiner’s findings in the Answer and Final Office Action as our own, except as to those findings that we expressly overturn or set aside in the Analysis that follows.

ANALYSIS

Appellant argues independent claims 1 and 11 together as a group with respect to the first and second issues. (App. Br. 5, 8-9, 12-13.) Appellant does not separately argue dependent claims 3, 4, 6-10, 13, 14, and

16-20. (*See* App. Br. 5-13, generally.) Appellant argues independent claims 21, 26, and 31 together as a group with respect to the first, second, and fifth issues. (App. Br. 13-15; *see* App. Br. 5, 8-9, and 12.) Appellant does not separately argue dependent claims 22-25, 27-30, and 32-35. (*See* App. Br. 5-13, generally.) Appellant also argues claims 2 and 12 together as a group with respect to the third issue (App. Br. 9-10) and argues claims 5 and 15 together with respect to the fourth issue (App. Br. 11-12). Therefore, we select claims 1, 2, 5, and 21 as representative of Appellant's arguments and groupings. 37 C.F.R. § 41.37(c)(1)(vii). *See In re Nielson*, 816 F.2d 1567, 1572 (Fed. Cir. 1987). We have considered only those arguments that Appellant has actually raised in the Briefs. Arguments that Appellant could have made but chose not to make in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Appellant has the opportunity on appeal to the BPAI to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) (citing *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)). The Examiner sets forth a detailed explanation of the obviousness rejection in the Examiner's Answer with respect to each of the claims (Ans. 3-18) and, in particular, claim 1 (Ans. 3-5, 14-18), claim 2 (Ans. 5, 18), claim 5 (Ans. 5) and claim 21 (Ans. 8-9). Therefore, we look to the Appellant's Brief to show error in the proffered findings and conclusions. *See Kahn*, 441 F.3d at 985-86.

*Arguments Concerning the Examiner's Rejection
of Representative Claim 1 Under § 103*

The Examiner rejects Appellant's independent claim 1 as being obvious over the combination of Smith and Averbuj. (Ans. 3-5, 14-18.)

Specifically, the Examiner submits that Smith teaches a method of creating network traffic replicating activities of telecommunications system users (citing Smith, Abstract), including: receiving a test script including a plurality of commands (citing Smith, col. 3, ll. 40-48, 59-67); invoking a script interpreter (citing Smith, col. 3, ll. 25-29; col. 4, ll. 14-18; Fig 1); and launching a thread (application thread) to execute the test script (citing Smith, col. 4, ll. 61-65). (Ans. 3.) The Examiner also submits that Averbuj teaches sequencers (protocol engines) that execute commands. (Ans. 4, 18 (citing Averbuj ¶¶ [0014], [0029], [0035], [0041], [0042], [0046]; Abstract).) The Examiner further explains that Smith and Averbuj are analogous art and that one of ordinary skill in the art would have been motivated to combine the references as proposed. (Ans. 4-5, 14-17.)

Appellant contends that the combination of Smith and Averbuj does not teach or suggest all the features of claim 1 (App. Br. 8-9, 12-13) and the Examiner has improperly combined the prior art references by resorting to hindsight (App. Br. 5-7).

Upon consideration of the evidence on this record and each of Appellant's contentions, we find that the preponderance of evidence on this record supports the Examiner's conclusion that the subject matter of Appellants' claim 1 is unpatentable over the combination of Smith and Averbuj. Accordingly, we sustain the Examiner's rejection of claim 1 for the reasons set forth in the Answer, which we incorporate herein by reference. (Ans. 3-5, 14-18.)

We recognize Appellant's contention, with respect to the first issue, that "the sequencers and test algorithms of Averbuj fail to teach or suggest

the protocol engine claimed” and that “[t]he term ‘protocol engine’ must be interpreted in view of the entirety of the claim as well as the specification . . . [and] [a]s such, the ‘protocol engine’ necessarily involves ‘network traffic’ and ‘test scripts.’” (App. Br. 8; *see* App. Br. 9; Reply Br. 3-4.)

However, Appellant’s claim simply recites “invoking a protocol engine for each of the commands in the test script,” so that “each protocol engine” is associated with a command of the test script (“has an associated command”), and “each protocol engine execut[es] its associated command.” (App. Br. 17, claim 1.) Appellant’s arguments are not commensurate with the scope of Appellant’s claim. Further, the disputed claim limitations merely require that the protocol engine has an associated test script command and executes the command. As explained by the Examiner, Smith describes a network test system receiving test scripts (Ans. 3) and Averbuj describes sequencers that each have an associated command and execute the command (Ans. 4, 18).

We also recognize Appellant’s contention, with respect to the second issue, that “Smith do[es] not teach or suggest the claimed application threads” but instead teaches “traditional operating system threads.” (App. Br. 12.)

Appellant asserts that the “application thread” term has a particular meaning as set out in the Specification — “[a]s set forth in the [S]pecification, an application thread is not a traditional operating system thread as it is lighter weight such that it requires ‘a smaller amount of network testing system resources to execute.’” (App. Br. 12 (*see* Spec. ¶ [0032]).)

We disagree and note that Appellant provides no persuasive evidence supporting these assertions or for the proposed particular definition.

Accordingly, we look to Appellant's Specification to inform our interpretation. We give claim terminology the "broadest reasonable interpretation consistent with the [S]pecification" in accordance with our mandate that "claim language should be read in light of the [S]pecification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004) (citations omitted). Appellant does not explicitly define, either in the Specification or the claim, the term "application thread" – rather, Appellant describes the application thread in terms of its function and its effect on system resources, explaining that:

Application threads are a lighter weight construct that require a smaller amount of network testing system resources to execute. That is, application threads require a smaller amount of memory and processor power to execute when compared with traditional threads. In addition, lighter weight threads require less communication between the thread and the operating system. As such, many more application threads may be executed in parallel or concurrently than traditional threads. Application threads may be implemented in many computer languages using various kinds of computer programming constructs. In one embodiment, the application threads are implemented using co-routines using the "C" programming language.

(Spec. ¶ [0032].) The disputed claim limitation also describes the application thread in functional terms, simply requiring that the application thread "execute the test script." (App. Br. 17, claim 1.) The claim does not recite or require that the application thread have any distinct structure or features. Thus, we simply interpret the "application thread" as executing the test script. Further, Appellant's proffered definition is merely an intended

use or purpose of the thread rather than a substantive definition of the terminology. As explained by the Examiner, Smith's threads perform this functionality (Ans. 3) – "each test script is preferably run on a specific channel and assigned to a specific thread of control" (Smith, col. 4, ll. 61-63). The claim limitation does not preclude Smith's threads.

Also, with respect to Appellant's combinability arguments (App. Br. 5-7), Smith and Averbuj are analogous art, in that both references teach complementary testing systems. Smith describes a method/system/apparatus for testing a telecommunication system. (Smith, col. 1, ll. 1-6; col. 3, ll. 10-12; Abstract.) Averbuj describes a testing system (Built-In Self-Test (BIST) system) for an electronic device in, for example, a communication system or computer network. (See Ans. 15; Averbuj, ¶¶ [0002], [0032]; Abstract.) We additionally note that the assignee in Averbuj is Qualcomm – a communication device design company, which further buttresses the Examiner's finding of analogous art. Further, the Examiner provides a rationale for combining Smith and Averbuj. (Ans. 4, 14-17.) We conclude that combining the teachings of Smith and Averbuj would have been tantamount to the predictable use of prior art elements and techniques according to their established functions – an obvious improvement. See *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007). We also conclude that the Examiner's articulated rationale:

a person of ordinary skill in the art would have readily recognized the desirability and advantages of modifying the system shown [in] Smith, to employ the features disclosed by Averbuj in order to offer the flexibility of allowing a variety of test algorithms to easily be defined and maintained centrally in the form of generalized commands, thereby eliminating the need to store common test algorithms in a distributed fashion

(Ans. 4 (citing Averbuj, ¶ [0015])) – i.e., improved efficiency and functionality – is based on “some rational underpinning to support the legal conclusion of obviousness.” *KSR*, 550 U.S. at 418 (quoting *Kahn*, 441 F.3d at 988).

Accordingly, we conclude, as did the Examiner, that the combination of Smith and Averbuj would have taught or fairly suggested the disputed features of: (1) “invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command, [and] each protocol engine execut[es] its associated command;” and (2) an “application thread” as recited in the claim. We find Appellant’s contrary arguments unpersuasive of error in the Examiner’s rejection for the reasons explained above. It follows that Appellant does not persuade us of error in the Examiner’s obviousness rejection of representative claim 1. Appellant also does not persuade us of error in the Examiner’s obviousness rejection of independent claim 11 and dependent claims 3, 4, 6-10, 13, 14, and 16-20 not separately argued with particularity (*supra*). Accordingly, we affirm the Examiner’s obviousness rejection of claims 1, 3, 4, 6-11, 13, 14, and 16-20.

*Arguments Concerning the Examiner’s Rejection
of Representative Claim 2 Under § 103*

The Examiner rejects Appellant’s dependent claim 2 as being obvious over the combination of Smith and Averbuj. (Ans. 5, 18.) Specifically, the Examiner submits that the commands in Smith’s test scripts simulate actions taken by user (Ans. 5, 18 (citing Smith, col. 2, ll. 19-32; col. 3, ll. 30-38)). Appellant contends that although “Smith teaches simulating the actions of a user[,] . . . the use of this teaching in the system constructed by the Examiner is impossible.” (App. Br. 9.)

Based on the record before us and Appellant's contentions, we find that the preponderance of evidence on this record supports the Examiner's conclusion that the subject matter of Appellant's claim 2 is unpatentable over the combination of Smith and Averbuj. Accordingly, we sustain the Examiner's rejection of claim 2 for the reasons set forth in the Answer (Ans. 5, 18), which we incorporate herein by reference.

Additionally, Appellant does not contest the Examiner's finding that Smith describes test scripts, and concedes that Smith teaches simulating user actions. (Ans. 5, 18; App. Br. 9-10.) Rather, Appellant argues the references separately and asserts that the sequencers and test algorithms (protocol engines) of Averbuj cannot be combined with the tests scripts of Smith. (App. Br. 9-10.) We find Appellant's contrary arguments unpersuasive of error in the Examiner's rejection for the reasons explained with respect to claim 1 (*supra*). Specifically, we find Appellant's combinability argument unavailing for the reasons set forth with respect to claim 1 (*supra*). Further, as explained with respect to claim 1, we note that Smith describes executing commands in test scripts and Averbuj describes sequencers executing commands/instructions. The combination of these features, contrary to Appellant's assertions, does "make logical sense" (App. Br. 10). Additionally, we also note that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *See In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986) (citing *In re Keller*, 642 F.2d 413, 425 (CCPA 1981)).

It follows that Appellant does not persuade us of error in the Examiner's obviousness rejection of representative claim 2 or claim 12, not

separately argued with particularity (*supra*). Accordingly, we affirm the Examiner's obviousness rejection of claims 2 and 12.

*Arguments Concerning the Examiner's Rejection
of Representative Claim 5 Under § 103*

The Examiner rejects Appellant's dependent claim 5 as being obvious over the combination of Smith and Averbuj and submits that Smith's test scripts cause network traffic to be produced. (Ans. 5 (citing Smith, col. 4, ll. 58-64; Abstract).) Appellant merely reiterates the arguments made with respect to claim 2 (*supra*) asserting that Smith's scripts cannot be invoked by Averbuj's sequencers (protocol engines) to produce network traffic. (App. Br. 11-12.)

We disagree for the reasons set out with respect to claim 2 (*supra*), and find Appellant's contrary arguments unpersuasive of error in the Examiner's rejection. It follows that Appellant does not persuade us of error in the Examiner's obviousness rejection of representative claim 5 or claim 15, not separately argued with particularity (*supra*). Accordingly, we affirm the Examiner's obviousness rejection of claims 5 and 15.

*Arguments Concerning the Examiner's Rejection
of Representative Claim 21 Under § 103*

The Examiner rejects Appellant's independent claim 21 as being obvious over the combination of Smith and Averbuj and submits that Smith describes script interpreter units in a user space, an application thread operating in a user space, and an operating system in an operating system space. (Ans. 8-9 (citing Smith, col. 3, ll. 49-67; col. 4, ll. 44-65).) The Examiner also submits that Averbuj describes protocol engines in a user space. (Ans. 8-9 (citing Averbuj, ¶¶ [0035], [0041], [0042]).) Appellant

contends that none of the references (Smith or Averbuj) describe a “user space” or an “operating system space,” much less individual elements in these spaces – e.g., protocol engines in user space – as recited in the claim. (App. Br. 13-15.)

Based on the record before us and Appellant’s contentions, we find that the preponderance of evidence on this record supports the Examiner’s conclusion that the subject matter of Appellant’s claim 21 is unpatentable over the combination of Smith and Averbuj. Accordingly, we sustain the Examiner’s rejection of claim 21 for the reasons set forth in the Answer (Ans. 8-9), which we incorporate herein by reference.

We provide the following for emphasis only. Appellant’s claim does not distinguish between “user space” and “operating system space.” In particular, the claim does not recite or require that these “spaces” be separate from one another. Indeed, the claim does not recite utilizing these spaces in any manner to do anything. As such, the “user space” and “operating system space” constitute non-functional descriptive material, which merely correspond to names or labels for the “space” (i.e., the portion or partition of system memory). How the spaces may be named or labeled does not functionally change the elements (e.g., the script interpreter units, application threads, protocol engines, and operating system) or the interaction among the elements, which is actually not recited in the claim. Merely labeling a portion of memory as a “user space” or a “operating system space,” as opposed to some other unique identifier, does not further limit the claimed invention either functionally or structurally. Thus, the label of the “spaces” represents

non-functional descriptive material, which “does not lend patentability to an otherwise unpatentable computer-implemented product or process.” *Ex parte Nehls*, 88 USPQ2d 1883, 1889 (BPAI 2008) (precedential). *See Ex parte Curry*, 84 USPQ2d 1272, 1274 (BPAI 2005) (informative) (Fed. Cir. Appeal No. 2006-1003), *aff’d*, (Rule 36) (June 12, 2006) (“wellness-related” data in databases and communicated on distributed network did not functionally change either the data storage system or the communication system used in the claimed method). *See also In re Ngai*, 367 F.3d 1336, 1339 (Fed. Cir. 2004); *Nehls*, 88 USPQ2d at 1887-90 (discussing non-functional descriptive material).

We also note that our discussion of claim 1 (*supra*) is equally applicable to claims 21-35.

It follows that Appellant does not persuade us of error in the Examiner’s obviousness rejection of representative claim 21 or independent claims 26 and 31 or dependent claims 22-25, 27-30, and 32-35 not separately argued with particularity (*supra*). Accordingly, we affirm the Examiner’s obviousness rejection of claims 21-35.

CONCLUSION OF LAW

Appellant has not shown that the Examiner erred in rejecting claims 1-35 under 35 U.S.C. § 103(a).

DECISION

We affirm the Examiner’s rejection of claims 1-35 under 35 U.S.C. § 103(a).

Appeal 2009-010841
Application 10/795,923

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

peb